

1. (Amended) A method for testing a computer program comprising the steps of:

parsing a source code of the computer program to identify functions in the source code;

responsive to the identified functions, generating stubs for the source code;

instrumenting the parsed source code with the generated stubs;

compiling the instrumented code;

testing the compiled code; and

reporting test results.

2. (Amended) The method of claim 1 wherein, the step of generating stubs comprises generating source code for replacing the name of externally called functions within the source code with the name of specific functions with same signature as the externally called functions.

3. The method of claim 2 wherein, the specific functions are one or more of predetermined functions and predetermined stubs.

4. The method of claim 3 wherein, the predetermined functions and stubs are automatically generated.

5. The method of claim 4 wherein, the step of automatically generating functions and stubs comprises automatically generating arguments to the functions and automatically initializing class members.

6. The method of claim 2 wherein, the specific functions are user-specified functions.

A² 7. (Amended) The method of claim 6 wherein, the user-specified functions are specified within a GUI.

8. The method of claim 1 further comprising the steps of breaking down the computer program into smaller components before compiling and testing the smaller components individually.

9. The method of claim 8 wherein, the smaller components are of the type of one or more of name space, class, function, and objects.

A³ 10. (Amended) The method of claim 1 wherein, the step of generating stubs comprises reconstructing a class by removing the source code that is not related to the class.

glb 11. (Amended) The method of claim 1 wherein, the step of generating stubs comprises reconstructing a class by ignoring the source code that is not related to the class.

12. The method of claim 2 further comprising maintaining a list of related functions to be replaced for each function under test.

13. The method of claim 1 further comprising monitoring test coverage of the computer program.

A⁴ 14. (Amended) The method of claim 13 further comprising displaying the monitored test coverage in a GUI as the test progresses.

15. The method of claim 1 further comprising the steps of defining a specific behavior when a function within the source code is called; storing the defined information; compiling the defined

information as a separate object; and linking the compiled object to the code.

16. The method of claim 1 wherein, the step of testing comprises of white-box testing.

17. The method of claim 1 wherein, the step of testing comprises of black-box testing.

18. The method of claim 1 wherein, the step of testing comprises of regression testing.

19. (Amended) A method for testing a computer program having a source code comprising the steps of:

AS parsing the source code of the computer program to identify a plurality of smaller components in the source code;

breaking down the source code into the plurality of smaller components;

based on the identified plurality of smaller components, generating stubs to replace some of the identified plurality of smaller components;

testing the plurality of smaller components individually; and

reporting test results.

20. The method of claim 19 wherein, the smaller components are of the type of one or more of name space, class, function, and objects.

21. The method of claim 19 further comprising replacing the name of externally called functions within the source code with the name

of specific functions with same signature as the externally called functions.

22. The method of claim 21 wherein, the specific functions are one or more of predetermined functions and predetermined stubs.

23. The method of claim 22 wherein, the predetermined functions and stubs are automatically generated.

24. The method of claim 23 wherein, the automatically generating functions and stubs comprises automatically generating arguments to the functions and automatically initializing class members.

25. The method of claim 21 wherein, the specific functions are user-specified functions.

26. The method of claim 19 further comprising monitoring test coverage of the computer program.

27. (Amended) The method of claim 26 further comprising displaying the monitored test coverage in a GUI as the test progresses.

28. The method of claim 19 further comprising the steps of defining a specific behavior when a function within the source code is called; storing the defined information; compiling the defined information as a separate object; and linking the compiled object to the code.

29. The method of claim 19 wherein, the step of testing comprises of white-box testing.

30. The method of claim 19 wherein, the step of testing comprises of black-box testing.

31. The method of claim 19 wherein, the step of testing comprises of regression testing.

32. (Amended) A system for testing a computer program comprising:

means for parsing a source code of the computer program to identify functions in the source code;

means for generating stubs for the source code responsive to the identified functions;

means for instrumenting the parsed source code with the generated stubs;

means for compiling the instrumented code;

means for testing the compiled code; and

means for reporting test results.

33. (Amended) The system of claim 32 wherein, the means for generating stubs comprises generating source code for replacing the name of externally called functions within the source code with the name of specific functions with same signature as the externally called functions.

34. The system of claim 32 further comprising means for breaking down the computer program into smaller components before compiling and means for testing the smaller components individually.

35. The system of claim 32 further comprising means for monitoring test coverage of the computer program.